



Self-Organization by Temporal Inhibition (SOTI)

P. MARTÍN-SMITH, F. J. PELAYO, E. ROS, and A. PRIETO

*Departamento de Arquitectura y Tecnología de Computadores, University of Granada, Spain.
e-mail: pmartin@atc.ugr.es*

Abstract. A model is presented for a neural network with competitive learning that demonstrates the self-organizing capabilities arising from the inclusion of a simple temporal inhibition mechanism within the neural units. This mechanism consists of the inhibition, for a certain time, of the neuron that generates an action potential; such a process is termed Post-Fire inhibition. The neural inhibition period, or degree of inhibition, and the way it is varied during the learning process, represents a decisive factor in the behaviour of the network, in addition to constituting the main basis for the exploitation of the model. Specifically, we show how Post-Fire inhibition is a simple mechanism that promotes the participation of and cooperation between the units comprising the network; it produces self-organized neural responses that reveal spatio-temporal characteristics of input data. Analysis of the inherent properties of the Post-Fire inhibition and the examples presented show its potential for applications such as vector quantization, clustering, pattern recognition, feature extraction and object segmentation. Finally, it should be noted that the Post-Fire inhibition mechanism is treated here as an efficient abstraction of biologically plausible mechanisms, which simplifies its implementation.

Key words: temporal inhibition, competitive learning, self-organizing maps, learning vector quantization

1. Introduction

The neural network model presented in this paper, in terms of its structure and learning processes, is based upon a competitive mechanism between the neural elements that comprise the network [1, 2]. This means that it is composed of sets of neurons with mutually-inhibiting connections that model lateral inhibition and are responsible for the competition between neurons. The proposed neural network model (SOTI) adds, at the neuron level, a particular temporal inhibition mechanism that we term Post-Fire inhibition, which consists of the self-inhibition that is subsequent to the emission of an action potential. For this study we chose a neuron model that was greatly simplified for the purposes of computer simulation. Nevertheless, this did not imply any loss of generality in relation to other definitions that might be considered more realistic from a biological point of view but which, for considerations of practical exploitation, would not justify their complexity. Thus the main focus of this study is to demonstrate the extent and applicability of the collective properties derived from a neural network that includes competition between its elements as a result of including the temporal self-inhibition mechanism. By this means it is possible to identify the collective behaviour patterns arising from

such a mechanism, which also represents a valuable simplification in terms of its physical implementation by electronic circuitry.

2. The Architecture and Basic Algorithm of the SOTI Model

The basic architecture of SOTI (which we call SOTIb), as shown in Figure 1, consists of a layer of neurons, each of which is connected to all the inputs, and of inhibitory connections between different neurons, which represent the basic competitive mechanism. The TEMPi modules represent the Post_Fire self-inhibition mechanism. Each of these modules inhibits the generation of an effective action potential during the presentation to the network of a number μ of vectors after the last effective action potential of the neuron U_i considered. This value of μ varies during the learning process, according to a time function ($\mu(t)$). This architecture thus obtains the winning neuron among those that are not inhibited by the TEMPi modules. It is also assumed that weight adaptation is performed in each iteration of the learning process on the winning neuron, which is the only one capable of generating an action potential. For the purposes of the simulation, given an iteration t and a stimulus vector $X(t)$, the calculation of the winning unit $U_g(t)$ is performed by means of expression (1).

$$U_g(t) \mid d(X(t), W_g(t)) = \min_j \{d(X(t), W_j(t)) \mid U_j \in CND(t)\} \quad (1)$$

$$CND(t) = \{U_j \mid (t - \tau_j(t)) > \mu(t)\}, \quad \text{with } j = 1, 2, \dots, Neu\}$$

where $d(\cdot)$ is the Euclidean distance, Neu represents the number of neural elements,

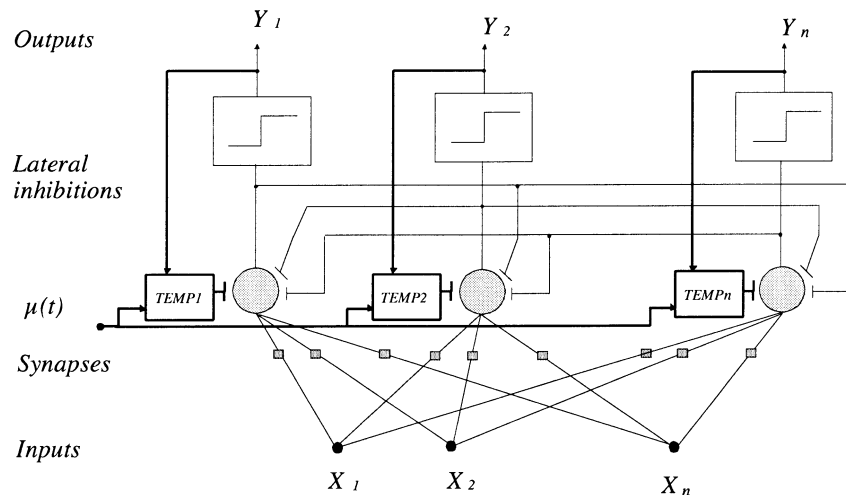


Figure 1. Basic architecture of the proposed SOTIb network. The model's main feature consists of including TEMPi modules that control the Post_Fire inhibition of the units.

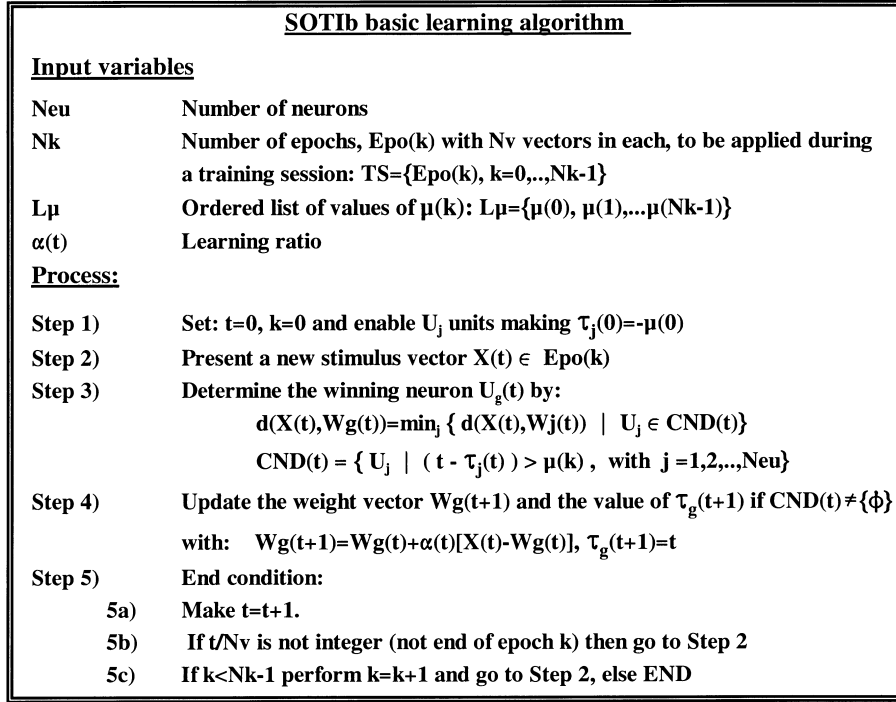


Figure 2. A simplified algorithm to simulate the SOTIb neural network.

W_j is the weight vector of unit U_j , $\tau_j(t)$ is the iteration in which U_j last won and $(t - \tau_j(t))$ represents the iterations that have occurred since that moment. Therefore, $CND(t)$ is the set of neurons that are not inhibited by the Post_Fire inhibition process. Expression (1) reduces to expression (2) when $\mu = 0$, in which $U_{g^*}(t)$ is the winning unit that is normally taken in a conventional vector quantization network VQ. Such a unit is termed the unconditional winning unit.

$$U_{g^*}(t) \mid d(X(t), W_{g^*}(t)) = \min_j \{ d(X(t), W_j(t)) \text{ for all } U_j \} \quad (2)$$

With respect to the adaptation of the weight vectors, we consider expression (3):

$$W_g(t+1) = W_g(t) + \alpha(t)(X(t) - W_g(t)) \quad (3)$$

The learning algorithm basically consists of iterating expressions (1) and (3), as summarized in Figure 2. In this algorithm, we have considered the presentation of the stimulus vectors as organized in epochs, $Epo(k)$, such that a constant value for the degree of Post_Fire inhibition is applied within each of them, i.e. $\mu(t) = \mu(k)$. This facilitates the definition of μ by using its dependence on the index k rather than on the iteration variable t .

3. Behaviour and Properties of SOTIb. Examples

In spite of the simplicity of the SOTIb algorithm, its behaviour is not trivial, and it presents characteristics that are very promising for exploitation in applications such as vector quantization, clustering, object segmentation and the extraction of spatio-temporal features. The characteristic properties of SOTIb are due to the Post_Fire inhibition mechanism, and in particular to the fact that this forces the neural elements to learn with stimulus vectors that are not limited to the standard regions of the Voronoi mosaics defined by the total quantity of neural elements comprising the network, but rather to the regions corresponding to the Voronoi mosaics defined by the subset of available units $CND(t)$ (see expression (1)), also taking into account that this subset changes dynamically during each iteration of the process. Indeed, it is a simple matter to verify that the degree of Post_Fire inhibition determines the number of elements Nnd of $CND(t)$ according to expression (4).

$$Nnd = Neu - \mu \quad (4)$$

Moreover, $CND(t)$ with a constant value of μ is formed by different elements at least until $\mu + 1$ iterations have taken place, as the unit that wins in iteration t ceases to be available during the next μ iterations. In other words:

$$\text{If } t_j > t_i \text{ with constant } \mu \text{ and } t_j - t_i < \mu \text{ then } CND(t_j) \neq CND(t_i) \quad (p1)$$

The dynamics of SOTIb is thus highly dependent on the way in which the degree of PostFire inhibition $\mu(k)$ is varied. As we shall see below, it is also dependent on the spatio-temporal structure of the stimulus vectors.

3.1. GENERAL BEHAVIOUR OF SOTIb WITH RESPECT TO $\mu(k)$

Concerning the definition of $\mu(k)$, let us first consider a decreasing linear dependence for μ in the following form:

$$\mu(k) = Neu - 1 - k \text{ where } k \text{ is the index of the epoch such that } k = 0, \dots, Neu - 1 \quad (5)$$

In this case, the number of elements of CND is given by $Nnd(k) = k + 1$. This means that in each epoch the number of units that are available to compete is increased by one. If $k = 0$ there is only one free neural element, which consists, for each iteration t , of the unit $U_{free}(t)$ that has just scaped from the Post_Fire inhibition. That is:

$$U_{free}(t) = U_g(t - \mu - 1) = \{U_j(t) \text{ such that } \tau_j(t) = t - \mu - 1\}$$

This unit is the only candidate to be the winning unit, i.e. $U_g(t) = U_{free}(t)$ and thus there is no competition. Moreover, during successive iterations, the $U_{free}(t)$ units follow a particular sequence to cover all the elements of the network. For $k = 1$, there are two non-inhibited neural elements that compete, according to a two-cell

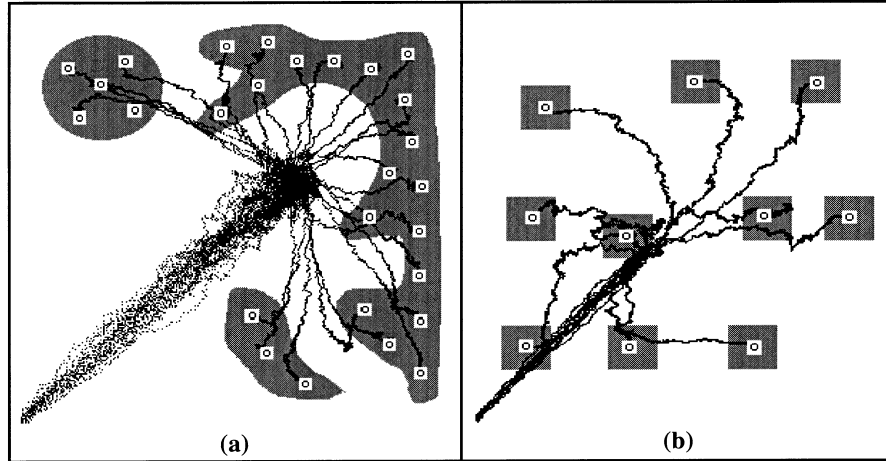


Figure 3. Execution of SOTIb that illustrate its effectiveness for vector quantization and clustering. Epochs with $Nv = 2000 \cdot p$ vectors and $\alpha = p \cdot Neu/Nv$ were used. In (a) we used the dependence of $\mu(k)$ from expression (6), $Neu = 27$ units and $p = 5$; in (b) we used the linear dependence from expression (5), $Neu = 10$ and $p = 2$.

Voronoi mosaic. By analogy with the previous case, the elements of $CND(t)$ change in each iteration according to property (p1). The process continues with decreasing values of μ until a zero value is reached during the epoch $k = Neu - 1$. At this moment SOTIb reduces to a basic competition scheme.

Overall, the Post_Fire inhibition process with a linearly decreasing $\mu(t)$ operates as a ‘drag’ mechanism that ‘forces’ the participation of the neural elements in the competition, with the result that these elements move towards the locations of the stimulus vectors. It represents a useful mechanism in terms of performing a vector quantization of the input space. Nevertheless, other decreasing dependences of $\mu(k)$ may obtain optimum results with an appreciable reduction in the execution time of the algorithm without having to apply the $Nk = Neu$ epochs corresponding to the different values of $\mu(k)$ defined by expression (5). Indeed, it is usually sufficient to have a subset of these to achieve a good trade-off between the duration of the algorithm and the final configuration of the weight vectors. Thus we have found [3] that the dependence of $\mu(k)$ given as

$$\mu(k+1) = 0.5 \cdot \delta(\mu(k)) - 1 \text{ with } \delta(n) = n + 1 \text{ if } n \text{ odd, and } \delta(n) = n + 2 \text{ if } n \text{ even} \quad (6)$$

using $\mu(k=0) = Neu - 1$, represents a suitable choice in most cases, which furthermore provides a lower limit for the number of different values of μ that are required.

Figure 3 illustrates the execution of SOTIb, demonstrating the “drag” mechanism of the weight vectors towards the stimulus regions and its utility in clustering and

vector quantization applications. The stimuli are randomly applied within the shaded areas in the figure. The white squares containing a circle indicate the weight vectors when execution of SOTIb has finalised, and the points reflect the trajectories of the weight vectors during training, with initial weights equal to zero. Figure 3a presents a generic example of vector quantization where μ varies according to expression (6). In this case, there is an optimal participation of all the neural elements in the vector quantization of the stimulus space, a situation which does not occur when a basic competitive learning scheme is used, in which in the end only one neural element wins for all the stimuli presented.

The example shown in Figure 3b uses the linear dependence of $\mu(k)$ given by expression (5). This shows the usefulness of SOTIb for clustering procedures and for optimal initialization of prototypes which could be used by other algorithms as a starting point, using a low number of weight vectors. Indeed, the literature contains many examples of vector quantization algorithms that require an initial number of prototypes that is compatible with the *a priori* probabilities of the classes that are implicit in the input vector space. One way to do it is to take as N initial prototypes, N randomly chosen vectors from the input space. However, to ensure this is achieved it is generally necessary to take a high value of N , which depends on the topology and the dimension of the vectors of the input space. In contrast, the example of Figure 3b contains 10 implicit classes (shaded squares) and 10 optimally-distributed weight vectors obtained by SOTIb (one per class), starting from weight vectors with a value of zero. The resulting weight distribution would have been highly improbable if 10 vectors had been randomly chosen from the input space.

3.2. BEHAVIOUR OF SOTIb AND SPATIO-TEMPORAL STRUCTURES OF THE STIMULI

With respect to the behaviour of SOTIb related to the spatio-temporal structures of the stimulus vectors, the Post_Fire inhibition provides important properties to the exploitation of the proposed network, as it is possible to extract characteristics from the input stimulus space in which the temporal context must be taken into account by the application to be solved. This aspect is normally ignored in conventional vector quantization algorithms, in which it is normal to apply the input vectors in a random fashion. Nevertheless, in many situations, information that is produced naturally is not random in nature; the brain processes and extracts characteristics that are dependent on the spatio-temporal structure of these natural ways by which stimuli are produced. Moreover, it is possible to induce the specific neural organization and behaviour that is desired by deliberately establishing the spatio-temporal structure of the input vectors to be processed. With the aim of characterizing and illustrating the behaviour of SOTIb in relation to the spatio-temporal structures of the stimuli, let us consider those defined in Figure 4, where an epoch is expressed in generic

<i>Spa. Temp Struc.</i>	<i>Label</i>	<i>Example</i>
Sequential between classes	<i>SC</i>	$Epo=SC\{[J_1C_1, J_2C_2]_p\}$
Trajectory	<i>TR</i>	$Epo=TR\{[J_1S_1, J_2S_2, \dots, J_nS_n]_p\}$
Random	<i>RND</i>	$Epo=RND\{[J_1C_1, J_2C_2]_p\}$

Figure 4. Spatio-temporal structures of stimuli and their notation.

terms as:

$$Epo = \text{Label of spatio--temporal structure } ([\text{basic structure}]_p) \quad (7)$$

where p is the number of times the basic structure (in square brackets) is repeated. The meaning of the spatio-temporal structure notation is clarified below, also including some simulation experiments of SOTIb with these spatio-temporal structures.

After this, we continue with the analysis and properties of SOTIb making use of the above-mentioned experiments.

Sequential between classes (*SC*)

In this case, the epoch consists of spatial groups or classes C_i , each of which contains $p \cdot J_i$ vectors. These vectors are applied to the SOTIb algorithm, in a temporal order that is defined by the class order specified in the definition of the epoch; for example, $Epo = \{J_1C_1, J_2C_2\}_{p=2}$ means that firstly, J_1 vectors from C_1 are presented, and then J_2 vectors from C_2 , after which the prior basic structure is repeated ($p = 2$), although this does not imply the presentation of the same vectors. Figure 5 shows a simulation with SOTIb using *SC* epochs. It uses a decreasing linear dependence for $\mu(k)$ (expression (5)) and illustrates the configurations of the weight vectors obtained for some values of μ .

Trajectory (*TR*)

This is analogous to the *SC* epoch, although in this case the stimulus vectors for each class S_i are aligned to form linear sections or curves. Furthermore, the vectors corresponding to each section S_i of the same epoch are required to be in an order that maintains a certain degree of spatial continuity between a vector and the one following it within the same section S_i . Examples of the natural production of epochs of the type *TR* are the sampled and multiplexed positions of a set of mobiles, the trajectories in multidimensional representations of speech signals, etc. Figure 6 shows a typical case, that of sampling the positions of a mobile with periodic move-

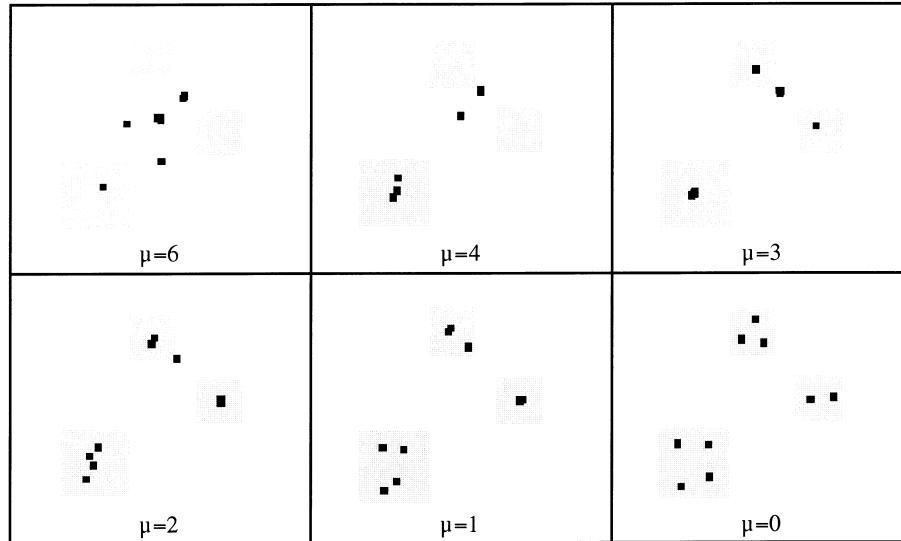


Figure 5. Example when SOTIb is executed with the epoch: $Epo(k) = SC[100C_1, 100C_2, 100C_3]_{p=5}$, $Nv = 1500$ and the parameters; $Neu = 9$, $\mu(k) = 8 - k$, $k = 0, \dots, 8$ and $\alpha = Nv/Neu$, showing the different configurations of weight vectors obtained when μ equals 6, 4, 3, 2, 1 and 0.

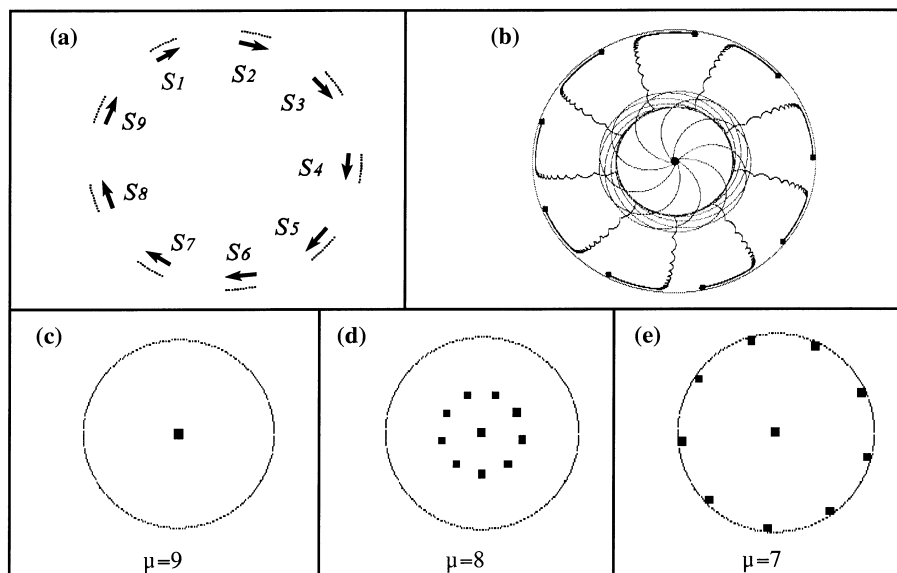


Figure 6. Execution of SOTIb using stimuli (x_1, x_2) generated continuously by applying $x_1 = \sin(\omega t)$ and $x_2 = \cos(\omega t)$ with $\omega = 2\pi t/8.97$. (a) 100 stimulus points and the 9 sections created by the subsampling effect. (b) The trajectories of the weight vectors. (c-e) The weight vectors with $\mu = 9, 8$ and 7 , using epochs with $Nv = 20000$, $\alpha = 0.01$, $Neu = 10$ and linearly decreasing $\mu(k)$.

ment at a frequency that is close to a multiple of the sampling frequency, thus producing epochs taking the form $TR[J_i S_i, i = 1, \dots, n]_p$ with $J_i = 1$.

The stimulus points (x_1, x_2) are generated by applying $x_1 = \sin(\omega t)$ and $x_2 = \cos(\omega t)$ with $\omega = 2\pi t/8.97$, which gives 9 S_i sections. Each S_i evolves following a circumference that starts at the positions indicated in Figure 6a. These, although they are spatially equal, are nevertheless out of phase with each other. The simulation illustrates how at $\mu = 9$ (Figure 6c) all the neural elements are concentrated in the centre and a sudden expansion of the weight vectors is produced at $\mu = 8$ (Figure 6d). Figure 6b shows the trajectories corresponding to this expansion process (inner circumferences). This phenomenon is a consequence of correlation $Ngp = \mu + 1 = 9$ winning units with nine stimuli from different sections, thus revealing a spatio-temporal structure $TR[1S_1, 1S_2, \dots, 1S_{Ngp}]$ for the input stimuli. Therefore, detection of such expansions would be useful for applications intended to search for this type of spatio-temporal structures in the natural production of stimuli.

Random (*RND*)

In this case, the epoch vectors are randomly presented. Note that it is possible to use the same vectors from an epoch that was initially defined as type *SC* or *TR* to obtain an *RND* epoch. For this purpose it is only necessary to establish a random presentation of the vectors to the neural network. It should be noted that this kind of application of stimuli, compared with *SC*, usually leads to a more uniform distribution of the weight vectors among classes. In this way, for example, if we reproduce the experiment illustrated in Figure 5 with an *RND* presentation of the stimuli, we obtain in $\mu = 0$ the result in Figure 7 instead of the one obtained in Figure 5 ($\mu = 0$) with *SC* presentation.

To return to the SOTIb analysis, a large part of its dynamics, particularly for the *SC* and *TR* spatio-temporal structures, becomes apparent by taking into account

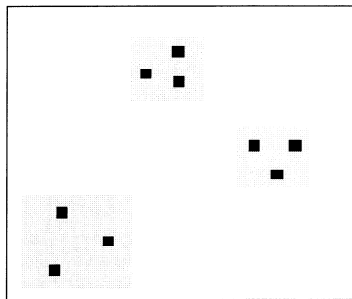


Figure 7. Weights obtained when $\mu = 0$, reproducing the experiment from Figure 5 with *RND*-type presentation.

that the Post-Fire inhibition induces a self-organization in the responses of neurons, which form groups that respond in a ‘forced’ cyclic fashion, following a fixed sequence, i.e. developing a rhythmic activity. We use the term *active group* to refer to each of these neuron sets, and the term *absolute maintained time of the group* to describe the number of consecutive iterations during which the rhythmic activity is maintained. Formally, an *active neural group* at instant t , $GP(t)$, is given by:

$$GP(t) = \{U_j \mid U_j \in CNI(t) \cup U_g(t)\} \quad (8)$$

$$CNI(t) = \{U_j \text{ such that } (t - \tau_j(t)) \leq \mu(t), \text{ with } j = 1, 2, \dots, Neu\}$$

i.e. the set of inhibited units $CNI(t)$ plus the winning unit at instant t , $U_g(t)$, which then forms part of $CNI(t+1)$ if μ is greater than zero. This set contains a quantity of elements that is given by:

$$Ngp = \mu(t) + 1 \quad (9)$$

The ‘forced’ cyclic response of the group GP_j appears when the group $GP(t) = GP_j$ does not change during an interval of iterations, which is defined by its absolute maintained time. Indeed, $GP(t)$ is formed by the $CNI(t) \cup U_g(t) = \{U_g(t - \mu), U_g(t - \mu + 1), \dots, U_g(t - 1), U_g(t)\}$. In the iteration $t + 1$ the unit $U_g(t - \mu)$ becomes the available unit, i.e. it does not belong to $CNI(t + 1)$, and therefore the group maintained time condition $GP(t) = GP(t + 1) = GP_j$ is equivalent to the condition $U_g(t - \mu) = U_g(t + 1)$, as expressed in the following property:

$$GP(t) = GP(t + 1) \quad \text{if and only if} \quad U_g(t - \mu) = U_g(t + 1) \quad (\text{p2})$$

This property, therefore, is responsible for inducing the fixed firing sequences of the winning units when the length of the sequence that is repeated coincides with Ngp . It also enables us to calculate, in a simple way, the Absolute Maintained Time (AMT) of a group GP , from expression:

$$AMT(GP(t) = GP_j) = T_f - T_i \text{ such that } \forall t \in [T_i, T_f - 1] \Rightarrow U_g(t - \mu) = U_g(t + 1),$$

$$\text{and } U_g(T_i - 1 - \mu) \neq U_g(T_i) \text{ and } U_g(T_f - 1 - \mu) \neq U_g(T_f) \quad (10)$$

with: $Gp_j = \{U_g(t - \mu), U_g(t - \mu + 1), \dots, U_g(t - 1), U_g(t)\}$

The particular firing sequence of the neurons in a particular active group Gp_j is established during the first Ngp iterations that led to the group being activated, and is maintained as long as Gp_j remains unchanged. This, however, does not mean that the same order is upheld in a subsequent activation of Gp_j after this has been interrupted. Moreover, during the time in which the group is active, the firing frequency fc of its units is very precise, and given by: $fc = 1/(Tm \cdot Ngp)$, where $Ngp = \mu + 1$ and Tm is the sampling period of the stimulous vectors. An additional point to note is that, during the learning process, the different active groups

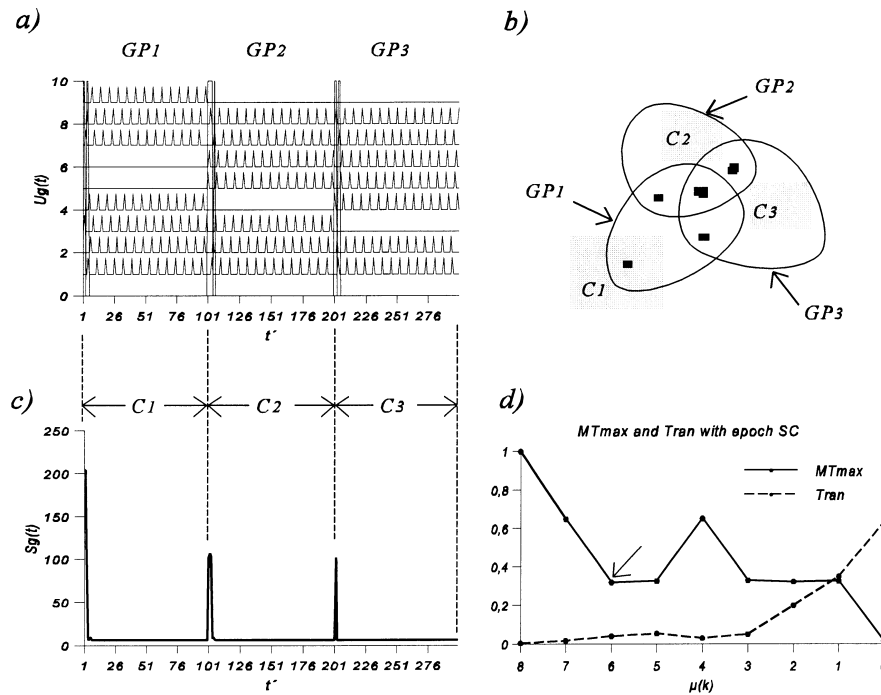


Figure 8. Experimental results corresponding to Figure 5. (a), (b) and (c) show results when $\mu = 6$. (a) Example of the self organization of the cyclic neural responses; (b) The weight vectors of the maintained active groups; (c) The signal $S_g(t) = t - \tau_g(t)$, useful for temporal segmentation of patterns; (d) The measures defined for MT_{max} and $Tran$ (expression (11)) versus the values of $\mu(k)$ that enable discrimination of the SC type presentation from RND (see Figure 9).

may have elements in common and differ only in certain units. The formation of groups with significant AMT is apparent when we use spatio-temporal structures of the type $SC = [J_i C_i, i = 1, \dots, n]$ with $J_i \gg Neu$ and $n < Neu$, as is the case of the simulation in Figure 5. As $\mu(k)$ decreases, active groups are formed with units that become more and more specialized in the different C_i classes. During the iterations in which a group is maintained, its neural elements learn cyclically with vectors $X(t)$ that are located in the classes that activate it; thus, the weight vectors tend to become grouped around their centres of gravity, taking into account that the weight vectors of the neural elements that are common to different active groups will be grouped in areas between the classes that activate such groups. The longer the AMT , the more efficient is this grouping mechanism. The experiment illustrated in Figure 5 clearly shows this phenomenon, while Figure 8 provides an example of the self organization of cyclic neural responses (Figures 8a,b), together with other results that are discussed below. Figure 8a illustrates the neural responses obtained

in iterations $t' = 1, \dots, 300$, corresponding to the last 300 iterations with $\mu(k) = 6$ from the experiment shown in Figure 5.

Note the formation of the active groups, with significant AMT , $GP_1 = [1, 2, 3, 4, 7, 8, 9]$, $GP_2 = [1, 2, 3, 5, 6, 7, 8]$ and $GP_3 = [1, 2, 4, 5, 6, 7, 8]$, which respond in a specific way to the respective classes C_1 , C_2 and C_3 , despite having just one neural element, unit 9, that responds specifically to a single class, while the others respond to more than one class. The figure also shows, with vertical lines, the iterations in which the active groups change, which is when: $U_g(t - \mu)U_g(t + 1)$ (see property (p2)). Figure 8c represents the signal $Sg(t) = \tau_g(t)$, i.e. the time elapsed since the last occasion when the unit $U_g(t)$ won; as can be seen, the peaks indicate the start of the presentation of a new class of stimuli. These peaks are caused by units that cease to respond in a specific way to one or more classes and that do respond in a specific way to another class or classes. For example, the peak on $t' \approx 1$, with $Sg(t) \approx 200$, is caused by neuron 9, as this neuron does not respond to the stimuli of two consecutive classes (Figure 8a) and thus remains inactive during approximately 200 iterations. In an analogous way, the second peak ($t' \approx 101$) is caused by neurons 5 and 6, while the third ($t' \approx 201$) is caused by neuron 4. Therefore, identification of the groups with an appreciable absolute maintained time, the iterations in which the active groups change and identification of the signal $Sg(t)$ are all useful in pattern segmentation when the natural production of stimuli is expected to be of the type SC with $J_i \gg Neu$. On the basis of these concepts, promising results in phoneme segmentation are currently being obtained. Other useful measurements in the exploration of active groups and in the discrimination of the type of spatio-temporal structure of the stimuli are defined by:

$$MT_{max}(k) = \max_j \left\{ \frac{AMT(GP_j)}{Nv/p} \right\} \quad Tran(k) = \frac{N_{change}}{Nv} \quad (11)$$

where Nv is the number of vectors in epoch k , p is the number of times the basic structure is repeated and N_{change} is the number of changes on the active groups. Figure 8d provides an example of these measurements for the experiment shown in Figure 5, where $Nv/p = \sum_i J_i = 300$ and $J_i = 100$. It presents stepped MT_{max} values that reveal a SC stimulus structure. Thus the MT_{max} values that are close to $J_i/300 = 1/3$ and $2J_i/300 = 2.3$ indicate the formation of active groups that respond hierarchically to 2 and 1 classes. On the other hand, from $\mu < 3$ there is an increasing number of changes of active groups, $Tran$. Thus, class C_1 captures 4 neural elements (see Figure 5 for $\mu = 3, 2, 1$ and 0) such that with $\mu = 2$ the active groups must contain $Ngp = \mu + 1 = 3$ elements, and so the random presentation of the stimulus vectors in class C_1 has the possibility of randomly activating any group of 3 neurons formed from the 4 that are located within class C_1 . By this process, the number of changes on the active groups increases and the random production of intraclass stimuli can be detected. In an analogous fashion, when $\mu = 2$, the C_2 class captures 3 neurons and the $Tran$ value increases when there is a greater number of groups that can be randomly activated with the C_1 class.

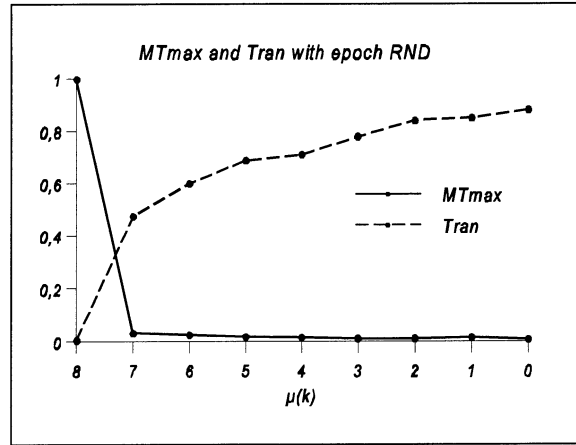


Figure 9. Measurements for MT_{max} and $Tran$ obtained by reproducing the experiment from Figure 5 with *RND* epochs.

These measurements enable us to obtain relevant information about the spatio-temporal structure of the stimuli and to determine whether a particular production of stimuli is random or has an intrinsic spatio-temporal structure that can be assimilated to type *SC* or *TR*. To demonstrate such a discrimination, Figure 9 gives the results of MT_{max} and $Tran$ obtained from the experiment described in Figure 5 when *RND*-type epochs are applied. The differences from the results given in Figure 8d are evident; it can be seen immediately, i.e. from $\mu < Neu - 1 = 8$, that the behaviour of the stimuli is random, in accordance with the low values of MT_{max} and the increasing $Tran$ values. When the stimuli are type *SC* or *TR* with $\sum_i J_i$ comparable to the number of neurons, the formation of groups with significant *AMT* is indicative of correlations that have been produced between the sequences of winning neurons in these groups and the presentation sequence of different classes of input vectors.

This fact is clearly apparent when $\sum_i J_i = \mu + 1$, as the neural elements are divided into subsets of neurons where each subset learns with a specific class of stimulus vectors. When this occurs, we observe sudden expansions of the weight vectors, such as when $\mu = 8$ in the experiment described in Figure 6. One important aspect to note is that the desired correlations may be induced, by deliberately establishing a dependence on $\mu(k)$ and a spatio-temporal structure of the stimuli that induces the learning of known neurons with particular classes of stimuli. This process is termed *induced supervised learning*. Figure 10 illustrates a simple example of this learning, where the epochs are prepared as follows: $Epo(k) = SC[J_i C_i$ with $J_i = J = 3$, $i = 1, \dots, 5]_p$; number of neural elements $Neu = \sum_i J_i = 15$; initial values $\tau_i(0) = i - 1 - Neu$ with $i = 1, \dots, Neu$ and the list $L\mu = \{Neu - 1 = 14, J - 2 = 1, J - 3 = 0\}$ to induce sets of J neurons per class such that the units U_s with

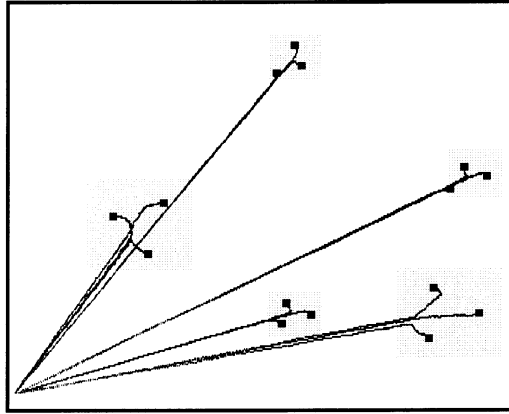


Figure 10. Induced supervised learning. Epochs are deliberately prepared to direct the learning process, such that 3 neurons are dedicated to each class. It is possible to identify, *a priori*, which neurons will learn with the vectors from a particular class.

$s \in [J \cdot (z - 1) + 1, J \cdot z]$ learn with vectors of the class C_z . This figure shows how the weight vectors are directed, from the start, to their respective classes and how they subsequently evolve in the local quantization of these classes.

4. Discussion

This paper presents a neural network model with competitive learning based on the temporal inhibition of its elements for a controllable time after the emission of action potentials. Taking into account the behaviour described in Section 3 and in [3], the applicability of the basic learning algorithm to typical applications of vector quantization is evident; it provides an efficient mechanism to ‘drag’ the weight vectors towards the regions where input stimuli are produced (Figures 3, 5, 6). Naturally, there exist other neural network models and VQ algorithms that achieve this result, but these are more complex than SOTIb, both from the computational viewpoint and in terms of their physical implementation. For example, in the cases of Self-Organizing Maps [4] and Growing Cell Structures [5], a neighbourhood of neural elements is defined for each winning unit, which requires the adaptation not only of the winning neuron but also of all those in its neighbourhood. These requirements do not exist with SOTIb. Another example is the case of the algorithms developed in [6], where learning must be carried out for sets of neurons that have to be ordered previously according to the distance of the weight vectors from the input vector presented during each iteration. Many other algorithms could be cited to demonstrate the relative simplicity of SOTIb. Moreover, we have stressed the fact that the Post.Fire inhibition represents a simple mechanism which, in addition to being biologically plausible, produces self-organized neural responses (Figure 8)

with a high information content. In this sense, the properties of the temporal inhibition mechanism, as shown in Section 3, provide the basis for exploiting the SOTIb model. The dynamics of the formation of active groups and their dependence on the degree of Post-Fire inhibition, as well as the ability to detect such groups and obtain measures of their absolute maintained time and changes of the active groups, justify the applicability of SOTIb to processes such as the extraction of intrinsic spatio-temporal features of stimuli (Figures 6, 8), pattern segmentation (Figures 8a,c) and induced supervised learning (Figure 10). The Post-Fire inhibition mechanism here described is also useful in a supervised learning context [3, 7].

Acknowledgements

This work has been partially supported by the Spanish CICYT Project TAP97-1166.

References

1. Rumelhart, D. E. and Zipser, D.: Feature discovery by competitive learning. In: D. E. Rumelhart and J. L. McClelland (eds), *Parallel Distributed Processing*, Vol. I, MIT Press, 1986, pp. 151–193.
2. Prieto, A., Martín-Smith, P., Merelo, J. J., Pelayo, F. J., Ortega, J., Fernández, F. J. and Pino, B.: Simulation and hardware implementation of competitive learning neural networks. *Lecture Notes in Physics* **368**, Springer-Verlag, 1990, 189–204.
3. Martín-Smith, P.: Un nuevo modelo de red neuronal con aprendizaje competitivo, PhD Thesis, University of Granada, Spain, November, 1997.
4. Kohonen, T.: *Self-Organization Maps*, Springer-Verlag, 1995.
5. Fritzke, B.: Growing cell structures – a self-organizing network for unsupervised and supervised learning. *Neural Network* **7**(9) (1994), 1441–1460.
6. Martinetz, T. M. and Schulten, K. J.: A neural-gas network learns topologies. In: T. Kohonen, K. Mäkisara, O. Simula, J. Kangas, eds. *Artificial Neural Networks*, North-Holland, Amsterdam, 1991, pp. 397–402.
7. Martín-Smith, P., Pelayo, F. J., Ros, E. and Prieto, A.: Supervised VQ learning based on temporal inhibition. *Lecture Notes in C.S.* **1606**, Proc. of IWANN'99. Springer, 1999, pp. 610–620.